

# Reduced Kernel Regression for Fast Classification

Fabian Hoti, Lasse Holmström  
Rolf Nevanlinna Institute, University of Helsinki, Finland

**Abstract:** We consider pattern classification using a weighted sum of normalized kernel functions. Such schemes can be viewed as estimates of class *a posteriori* probabilities. We apply this regression method successfully to two real life pattern recognition problems.

## 1 Introduction

Pattern recognition has applications in various fields such as speech recognition and classification of handwritten characters. Discrimination techniques used in pattern recognition are divided into parametric and non-parametric according to their structures. In this paper we introduce a method that lies somewhere between these two categories. First we will introduce the basics of classification and the theoretically optimal Bayes classifier. Next we will use a kernel regression method to approximate the Bayes classifier and finally we will reduce its complexity considerably. The resulting method can then be viewed either as a local parametric method or as a non-parametric radial basis function expansion familiar from neural network studies.

The performance of the method is tested with two different real world data sets and the results are compared to those gained from popular classifiers [2]. It appears that our method enables us to combine the flexibility of non-parametric methods with the speed of parametric methods.

## 2 Statistical Classification

Let  $x = [x_1, \dots, x_d]^T \in \mathcal{R}^d$ , be a pattern vector, a multidimensional measurement, taken from an object that belongs to one of  $c$  different classes. Given a pattern vector  $x$ , a classification problem is to guess from which class  $j \in \{1, \dots, c\}$  the measurement originated. A classifier can now be regarded as a function  $g : \mathcal{R}^d \rightarrow \{1, \dots, c\}$ . Note that pattern vectors from different classes tend to overlap.

In the sense of probability theory the whole observation is a  $d + 1$  dimensional random vector  $[X^T, J]^T$ . Let  $P_j = P(J = j)$  be the *a priori* probability and  $f_j$  the probability density of class  $j$ . Now the density of the pooled data is  $f = \sum_{j=1}^c P_j f_j$  and the *a posteriori* probability of class  $j$  conditional on  $X = x$  is

$$q_j(x) = \begin{cases} P_j f_j(x)/f(x), & \text{if } f(x) \neq 0 \\ 0, & \text{if } f(x) = 0, \end{cases} \quad (1)$$

The optimal classifier in the sense that it minimizes the probability of misclassification is called the Bayes classifier. It can be shown to be given by

$$q_{BAYES}(x) = \operatorname{argmax}_{j=1,\dots,c} q_j(x). \quad (2)$$

In case the choice is not unique, it can be made freely among the classes with the largest value of  $q_j(x)$ . In practice classifiers are constructed either by combining (1) with (2) and estimating the class densities  $f_j$  or by using some regression technique to directly estimate the *a posteriori* probabilities  $q_j$  from the given data.

### 3 Reduced Kernel Regression

Let  $\{(X_1, J_1), \dots, (X_n, J_n)\}$  be a sample,  $X_i$  a  $d$ -dimensional pattern vector and  $J_i$  its corresponding class label. We estimate the class densities with the *kernel estimator* [6]

$$\hat{f}_j(x) = \frac{1}{n_j} \sum_{i=1}^n Y_i^j K_h(x - X_i), \quad (3)$$

where  $n_j = \#\{i : J_i = j\}$ ,  $Y_i^j$  is 1 if  $J_i = j$  and 0 otherwise. Further,  $K_h(x) = h^{-d}K(x/h)$ , where  $K$  is a kernel, a nonnegative function, which integrates to one and is symmetric about the origin, and  $h$  is a smoothing parameter. We estimate the *a priori* probabilities with  $\hat{P}_j = n_j/n$ . Finally, we derive an estimate for the  $j$ 'th class *a posteriori* probability as follows

$$q_j(x) = \frac{P_j f_j(x)}{f(x)} \approx \frac{\hat{P}_j \hat{f}_j(x)}{\sum_{k=1}^c \hat{P}_k \hat{f}_k(x)} \quad (4)$$

$$= \frac{\frac{n_j}{n} \frac{1}{n_j} \sum_{i=1}^n Y_i^j K_h(x - X_i)}{\sum_{k=1}^c \frac{n_k}{n} \frac{1}{n_k} \sum_{r=1}^n Y_r^k K_h(x - X_r)} \quad (5)$$

$$= \frac{\sum_{i=1}^n Y_i^j K_h(x - X_i)}{\sum_{r=1}^n K_h(x - X_r)} = \hat{q}_j(x). \quad (6)$$

The result happens to be identical with the Nadaraya—Watson kernel regression estimator [8]. Note that the estimate is a weighted sum over  $Y_i^j$  and can be written as

$$\hat{q}_j(x) = \sum_{i=1}^n Y_i^j w_{i,h}(x), \quad (7)$$

where  $w_{i,h}(x) = K_h(x - X_i) / \sum_{r=1}^n K_h(x - X_r)$  and  $\sum_{i=1}^n w_{i,h}(x) = 1$ . We further improve the flexibility of the estimator by replacing the constants  $Y_i^j$  with locally

fitted functions, for example polynomials. In this study we consider the use of first-order polynomials

$$\hat{q}_j(x; a, b) = \sum_{i=1}^n (a_i^j + x^T b_i^j) w_{i,h}(x), \quad (8)$$

where  $a = (a_i^j), b = (b_i^j), i = 1, \dots, n, j = 1, \dots, c$  and  $a_i^j \in \mathcal{R}, b_i^j \in \mathcal{R}^d$ . The fitting is done in the mean squared error sense

$$\frac{1}{n} \sum_{i=1}^n \|\hat{q}(X_i; a, b) - Y_i\|^2 = \min_{a,b}, \quad (9)$$

where  $\hat{q} = (\hat{q}_1, \dots, \hat{q}_c)$  and  $Y_i = (Y_i^1, \dots, Y_i^c)$ . Note that (9) approximates the mathematical expectation  $E[\|\hat{q}(X) - Y\|^2]$  which attains its smallest value when  $\hat{q}(x) = E[Y|X = x] = q(x)$ . The achieved scheme is a very flexible estimator, but as the dimension of the data increases it is due to run into trouble. The estimator suffers from the curse of dimensionality [5]: in higher dimensions the amount of data needed to get good estimates increases rapidly.

We attack the problem of dimensionality by adapting some methods from neural network studies. The Radial basis function expansion of Moody and Darken [4] has the form

$$\hat{q}_j(x) = \sum_{r=1}^M \frac{a_r^j K(\frac{x-m_r}{h})}{\sum_{s=1}^M K(\frac{x-m_s}{h})}. \quad (10)$$

Here  $M$  kernels with centers at locations  $m_r$  are used to produce a weighted sum of the constants  $a_r^j$ . This scheme is fast to use because its complexity depends on  $M$  which is chosen to be considerably smaller than  $n$ . Now by combining the flexibility of local fitting in scheme (8) and the speed achieved in scheme (10) we get the Reduced Kernel Regression scheme

$$\hat{q}_j(x) = \sum_{r=1}^M (a_r^j + x^T b_r^j) v_r(x), \quad (11)$$

where  $v_r(x) = K(\frac{x-m_r}{h_r}) / \sum_{s=1}^M K(\frac{x-m_s}{h_s})$ , and  $K(x) = C \exp(-\|x\|^2)$ . Here the constants  $a_r^j \in \mathcal{R}$ , vectors  $b_r^j \in \mathcal{R}^d$ , kernel centers  $m_r$  and smoothing parameters  $h_r$  are chosen by the help of training data. As a fitting criterion we use the least-squared error (LSE). A scheme similar to (11) was proposed in [7]. In Figure 1 we demonstrate the use of this method by estimating a sine function from noisy data. Note that after the centers  $m_r$  and the smoothing parameters  $h_r$  are fixed, calculating the remaining parameters is a linear least-squares problem, which can be solved, e.g., by matrix pseudoinversion.

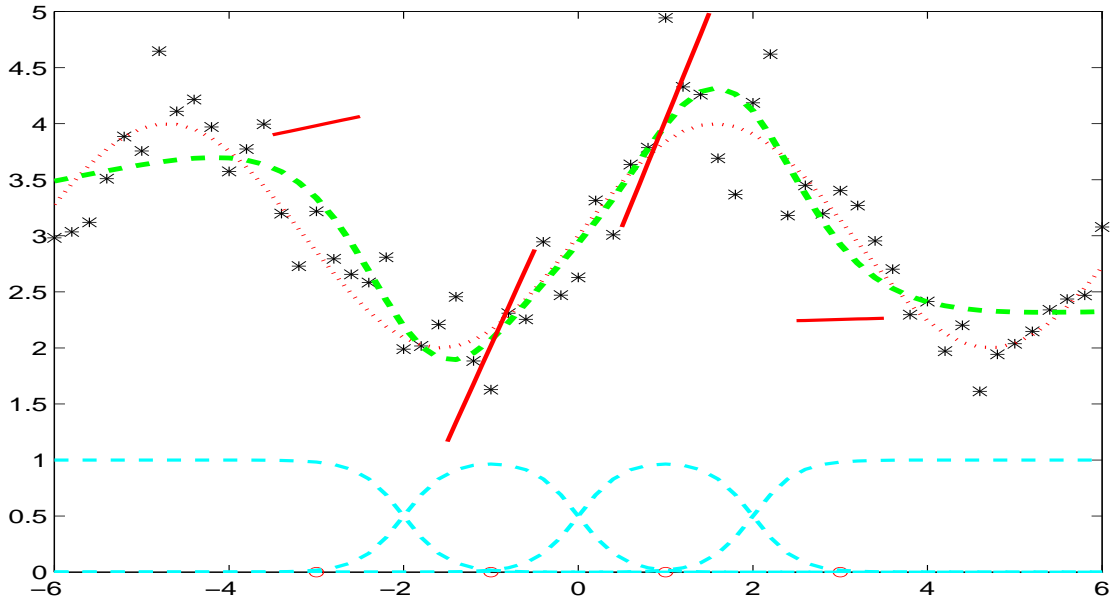


Figure 1: The sine function (dotted) with added noise (stars). Four weight functions  $w_k$  (dashed lines on the bottom) and their centers  $m_k$  (circles). The coefficient functions  $a_k + b_k^T x$  are drawn as a solid line about the centers  $m_k$  corresponding to them. The estimate is given by the dashed line.

## 4 Case Studies

We applied the Reduced Kernel Regression (RKR) scheme to two real life data sets. The same data have been used in an extensive comparison study of neural and statistical classifiers [2].

### 4.1 Phoneme recognition

The phoneme data were first used in the European ROARS ESPRIT project [1]. The aim of the project was to develop a pattern recognition system for spoken French. The data are composed of two classes in 5 dimensions. Class 1 corresponds to nasal French vowels and class 2 to oral French vowels. As seen in Figure 2 the data is non-normally distributed and has a rich internal structure. The data consisting of 5404 vectors, 3818 nasal and 1586 oral vowel patterns, are divided into equally sized training and test data sets.

After experimenting with different algorithms we adopted the following one.

1. Initialize  $M/2$  centers  $m_r$  to both classes with reference vectors of an  $M/2$ -unit Kohonen Self-organizing Map (SOM) [3] trained with training vectors from the associated class only.
2. Initialize  $h_r = h$ , where  $h$  is the average distance between all centers  $m_r$ .

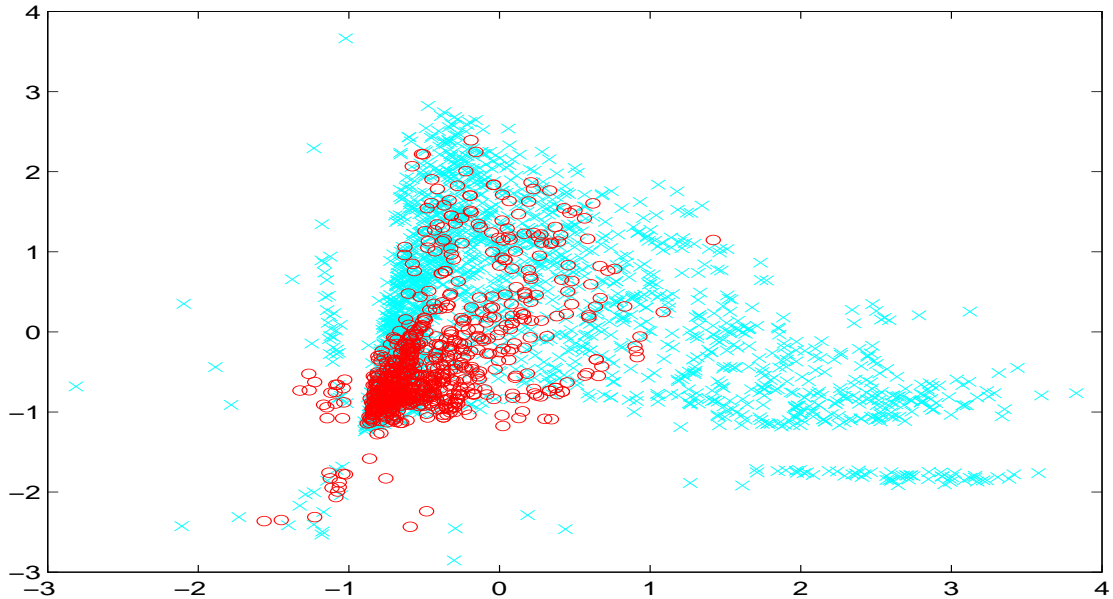


Figure 2: The first two components of the phoneme data. Class 1 (black circles) and class 2 (grey crosses).

3. Select the constants  $a_r^j = 1$ , when  $m_r$  is associated with class  $j$  and  $a_r^j = 0$  otherwise. Select the vectors  $b_r^j = 0$  always.
4. Use a minimization algorithm to optimize the centers  $m_r$  and the smoothing parameters  $h_r$ .

Notice that we use the SOM algorithm only for clustering and that any other clustering method could be used as well. Keeping in mind the main idea that  $M \ll n$  we chose  $M$  to be 200. We minimized the squared error using the steepest descent algorithm. The step length was chosen with parabolic interpolation. We determined the number of steps by using 10-fold cross-validation on the training data. Thus, the training data were divided into 10 parts, one part at a time was used as the test data and the rest were used for training. The result is the average over all 10 trials. The chosen number of steps was 100 (Figure 3), which gave the test data a classification error of 13.0%. This was calculated as an average over 10 independent runs of the SOM algorithm. (The SOM algorithm produces each time a different set of reference vectors.) The standard deviation of the mean was estimated to be 0.02%. In Table 1 we compare our result with some results obtained using other kernel based classifiers [2]. Local linear regression (LLR) and kernel discriminant analysis (KDA) are non-parametric methods and quadratic discriminant analysis (QDA) is a parametric method, which assumes that all classes are normally distributed. LLR estimates directly the *a posteriori* probabilities and KDA uses (3) to estimate the class densities.

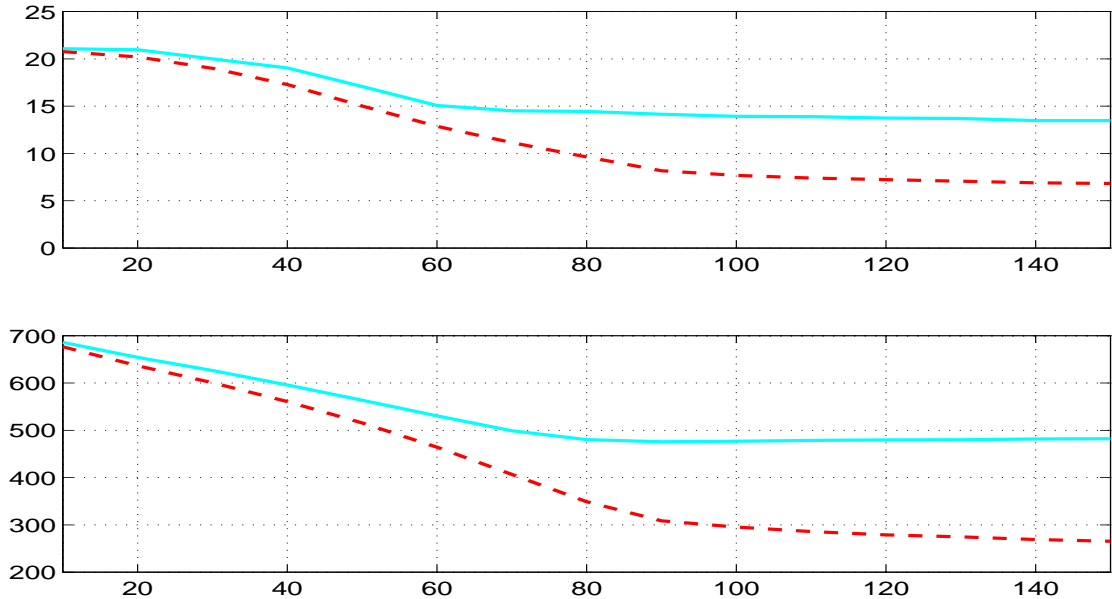


Figure 3: Determining the number of steps by 10-fold cross-validation. The upper plot shows the classification error of the training part (dashed) and the test part (solid). In the lower plot we have the value of the error function (LSE), respectively. Note that the values are means over all 10 cases and in the lower plot the test values have been scaled by the factor 9.

## 4.2 Handwritten digit recognition

The handwritten digit data [2] were collected on forms filled out by 894 randomly chosen Finnish people. Each person filled in two examples of each of the ten digits. The images were scanned in binary form with the resolution of 300 pixels per inch in both directions. The bounding boxes of the digits were normalized to  $32 \times 32$  pixels and the slant of the images was removed. The sample, 17880 1024-dimensional binary vectors, was then divided into two sets of equal size, one for training and one for testing. The binary vectors were transformed into 64-dimensional real vectors using the Karhunen-Loève (KL) transformation determined from the covariance matrix of the training data.

After the preprocessing the components of the vectors were roughly normally distributed. The algorithm we applied to the phoneme data did not work on these data. So after doing some more experiments we adopted the following algorithm.

1. Assign  $M$  centers  $m_k$  common to all classes with reference vectors of an  $M$ -unit SOM, trained using the whole training data.
2. Optimize  $a_k^j, b_k^j$  and  $h_k = h$  by 3-fold cross-validation on the training data. (Calculate for each value  $h$  the optimal parameters  $a_k^j$  and  $b_k^j$  by using matrix pseudoinversion.)

Method	centers	error	Method	dim	centers	error %
LLR	2702	12.9	LLR	36	8940	2.8
KDA	1909+793	11.3	KDA	36	$10 \times 894$	3.5
QDA	$2 \times 1$	21.7	QDA	47	$10 \times 1$	3.7
RKR	$2 \times 100$	13.0(.02)	RKR	30	25	3.5(0.001)

Table 1: The left table contains the results for the phoneme data and the right table for the handwritten digit data. In both tables we have included the method, the classification error and the number of centers, which describes the complexity of the method. In the digit case we have also given the amount of components used. In parentheses are the estimated standard deviations in 10 independent trials.

We also used 3-fold cross-validation to further reduce the pattern space dimension. The decision was to use 30 first KL-components. The available computing resources allowed us to use  $M = 25$  and the classification error was 3.5%. We did the same comparison as in the earlier case (Table 1).

## 5 Conclusions

We have derived the estimator (11), and applied it to two different pattern recognition tasks. Our aim was to handle successfully both cases with one method. The complexity of the phoneme data suggests that only non-parametric methods should be considered. On the other hand, the use of non-parametric methods, especially with the handwritten digits data, is computationally very heavy.

We succeeded to get good classification results in both cases and at the same time we were able to reduce the computational cost (measured as the number of kernels) considerably compared to the other non-parametric methods. We encourage the use of RKR as a universal method and especially in cases when fast classification is needed and the use of parametric methods is not possible.

## 6 Acknowledgements

The work of F. Hoti was partially supported by Grant 38122 from the Academy of Finland. The work of L. Holmström was partially supported by the National Science Foundation Grant DMS-9631351.

## 7 References

- [1] P. Alinat: Periodic progress report 4. Technical report, ROARS Project ESPRIT II-Number 5516, February 1993. Thomson report TS. AMS 93/S/EGS/NC/079.
- [2] L. Holmström/P. Koistinen/J. Laaksonen/E. Oja: Neural and statistical classifiers—taxonomy and two case studies. *IEEE Transactions on Neural Networks*, 8:5–17, 1997.
- [3] T. Kohonen: *Self-organizing Maps*. Springer-Verlag, 1995.
- [4] J. Moody/C. Darken: Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.
- [5] D.W. Scott: *Multivariate Density Estimation*. John Wiley & Sons, 1992.
- [6] B. W. Silverman: *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [7] K. Stokbro/D.K. Umberger/J.A. Hertz: Exploiting neurons with localized receptive fields to learn chaos. *Complex Systems*, 4:603–622, 1990.
- [8] M.P. Wand/M.C. Jones: *Kernel Smoothing*. Chapman & Hall, 1995.

MSc. Fabian Hoti  
Rolf Nevanlinna Institute  
P.O Box 4  
FIN-00014 University of Helsinki, Finland  
+358-9-191 22770 (Tel.)  
+358-9-191 22779 (Fax)  
Fabian.Hoti@RNI.Helsinki.Fi

Dr. Lasse Holmström  
Rolf Nevanlinna Institute  
P.O Box 4  
FIN-00014 University of Helsinki, Finland  
+358-9-191 22776 (Tel.)  
+358-9-191 22779 (Fax)  
Lasse.Holmstrom@RNI.Helsinki.Fi